1. Introduction

1.1. Objective

This project is trying to use YOLO to detect several different targets among 15000 images including biker, car, pedestrian, traffic Light, traffic Light-Green, traffic Light-Green Left, traffic Light-Red, traffic Light-Red Left, traffic Light-Yellow, traffic Light-Yellow Left, and truck in images. Those objects will be detected and the probilities of the detected objects will be provided from the YOLO algorithm. YOLOv3 will be used in this project because it is fast, accurate, stable and fully developed.

15000 images with 512*512 pixels from Udacity self driving car dataset in roboflow will be used to a pre-trained model which trained on big datasets, such as COCO dataset, VOC dataset. The pre-trained model will be used as an initialization of the training.

1.2. Significance

Autonomous driving will be the leading way of transportation in the future, because it lowers the risk of making mistakes, increases stability and satfy, provides passive protection beyond human cognitives. In the meanwhile, an automatic system with dedicatedly constructed planning strategies improves the efficiency of road planning, decision making, and so on so forth, liberating drivers while increasing confomtability. The current automonous driving systems relies highly on visual information to make decisions, e.g., detecting mutlple objects simultaneously, estimating the depth map of an image, semantically segment multiple instances. Another important aspect of autonomous driving is to make decisions in real time, as the lower the latency is, the safer the system will be. Therefore, recognizing multiple targets in real-time is highly demand on the decision making of self-driving systems.

This leads us to YOLO (you only look once) algorithm which has capaibility of detecting multiple objects in real time with reasionable performance. The Original YOLO paper demonstrates its ability on performing object detection on videos with 45 FPS, and even 150 FPS with a tiny version. At that time, the other state-of-the-art methods, e.g., Faster R-CNN is only able to process images at 7 FPS. The essential ideas of YOLO is that it provides a unified neural network to train and do inference instead of a multi-stage system inferring on neural network several times. The motivation to do so is that feeding data to a (convolutional) neural network is an extremely time-consuming process even on GPU. The R-CNN based methods are either running a neural network multiple times during inference, or running multiple neural networks to do regional proposals and detection, seperately. The inherent computation of selecting regional proposal is the key factor that R-CNN based method can not run inference in a fast matter. The millisecond-level inference enables YOLO to be way better than human cognitives whose reaction time is always at half second.

Furthermore, not only YOLO is fast, it also accurate in terms of the mean average precision (mAP). The original papers demonstrates its mAP is doubled compared to the other state-of-the-art work. For autonomous driving, precision is of equal importance as real-time compatibility. The higher accuracy the algorithm is able to achieve, the less risk the driver will face. For instance, it is disastrous for an autonomous driving system to fail detecting a pedestrian, a car in front of a driver. YOLO makes a big progress in real-time detection. It is the most popular and reliable way to detect the multiple targets in real-time based on the image information got from camera, that is also the reason why we choose this system as a main approach to the detection in the project. Conbination of YOLO, Lidar, Ridar, and HD(high-definition map) leads to a reliable, safty auto-driving system which will bring huge pleasure and convenience for humen.

2. Background

2.1. Current State of Research

Object detection is one of the most significant and hottest high-level vision tasks which enpowers a varities of applications in automous driving, surveillance, machine inspection, etc. The fundamental problem of object detection is to classifty potential objects within an image while localizing them simultaneously. The former task catergorized the objects within an image or a patch of an image, if pathc-based object detection methods considered. Classification is not an overwhelmingly intimating task, the convolutional neural network (CNN) revolutionizes the accuracy of object classification year-by-year, e.g., ImageNet[7], AlexNet[7], ResNet[8], etc. On the contrary, localization is comparatively harder to achieve. Even before the emergence of convolutional neural network, there are many state-of-the-art methods which always use hand-crafted features (e.g., spatial representation, local statistics) to represent an object. However, the hand-crafted features sometimes are biased in some cases, especially when occlusion exists. Ideally, features ought to be learned from the data itself. Therefore, we only focus on CNN-based methods in this paper. But notice that Transformer[9] and Mixer[10] are also another powerful architectures other than CNN, which imposes different mechanisms to represent spatial features.

At the very early stage of the research in object detection, people always constructed a two-stage systems: the first is to train a classifier based on sub-patches of images where a certain object is labelled; the second is to perform inference with the trained classifier at different scale (from coarse to fine) and aggregate the results at different scales as the final localization, because objects are always at different scales. One of the most widely-used methods is silding window at different scales, which sweeps overlapping patches of an image with a set of scaled windows and performs inference at each window. This method is computationally expensive in reality especially in the cases where the classifier is (convolutional) neural network. Another issue of this method is the ill-posedness of precise localization on deformable objects. The Overfeat method proposed by Sermanet et al[4] mitigates the inefficience in computation of the sliding window by applying the sliding window directly over the feature map of the CNN instead of over the input images. The key idea is that convolution can preserve the spatial information. Performing sliding window over the input image and feeding the CNN is equilvent to feeding the whole image to the CNN and performing sliding window over the feature map which can be done by another convolutional layer. This method, however, still needs to perform coarse-to-fine inference which means running the network several times. Girshick et al.[11], mitigate this issue by proposing the Regional proposal method named R-CNN. The essential idea to reduce the number of potential patches which contains objects by performing a preliminary segmentation over the raw image. By doing so, only the patches (~2k) potentially contain the objects will be feeding to the neural network to extract the feature and then train a support vector machine (SVM) to classify the image based on the extracted features. Even though this method reduces the potential patches containing objects to the network, it runs network for each patch which is time-consuming. The fast R-CNN[12] and faster R-CNN[13] aim to mitigate this issue by changing the way to do regional proposal. The fast R-CNN feeds the whole image in a CNN and perform regional proposal over the feature map meaning only one network is running for each image. The faster R-CNN proposed a regional proposal network (RPN) to do the regional proposal which essentially takes advantage of the fact that the extracted feature used for regional classification can also be used to do regional proposal.

YOLO (You Only Look Once) is a powerful new technology in computer vision which is proposed by R. Joseph[1] in 2015. It only applied one single neural network to the full image which improve the speed and bring the objects-detection to real-time. As its name, it can identify multiple specific objects in videos, live feeds, or images all at once.

2.2. Hypotheses & Questions to Answer

The accuracy used to be a weakness of YOLO, especially for small objects that appear in groups, such as flocks of birds[5]. However, it can be a trade-off between the speed and accuracy. In[3], it said the trade-off between speed and accuracy can be simply changed by the size of the model. The way that making the model both high speed and high accuracy will be one of questions need to be answer in the further research. In addition, the comparision and development for YOLOv1 to YOLOv5 will be summarized. The approachs and attepts of improving YOLO is another question need to be answer in the future work.

3. Research Plan

YOLOv3 will be used to detect multiple objects in this project. YOLOv3 is a fully developed, fast, and accurate real-time objects detection system. In mAP(mean average presion) measured at 0.5 IOU, YOLOv3 is on par with Focal Loss but about 4x faster[6]. Which inspired by ResNet and FPN (Feature-Pyramid Network) architectures, YOLO-V3 use a feature extractor called Darknet-53 (it has 52 convolutions) contains skip connections (like ResNet) and 3 prediction heads (like FPN) — each processing the image at a different spatial compression[3].

Since the training of YOLO is computational expensive, a pre-trained model which trained in big dataset with the multiple targets will be found in the first step. The online sources of YOLOv3 system will be used. Then, 80% of the 15000 images will be used to training the model, 10% will be used in the validation test. Mean average precision(mAP) will be used as the evaluation criterion. After that, I will try to use some different hyperparameters to train the model for several times until get a good mean average precision value. Finally, the last 10% of the dataset will be used to test the YOLO model. The output of mean average precision evaluate the accuracy of the model. From now to Nov 12, I will learn the way to achieve YOLOv3 its background knowledge and the development from YOLOv1 to YOLOv5. Then, it will take around 2 weeks to set up, debug, train, and test the model. Finally, one week will be left before the deadline to work on constructing the report. The timeline of the project is briefly shown in the following table:

Tasks	Timeline	State
Task1: Learn YOLOv3	Oct 20 – Nov 12	Ongoing
Task2: Research for the development of YOLOv1-v5	Oct 20 – Nov 12	Ongoing
Task2: Train and test YOLOv3	Nov 13 – Nov 24	Not start
Task3: Work on the report	Nov 25 – Dec 2	Not start

Table3.1

The timeline of the tasks may slightly change during the research.

References

[1] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).
[2] https://pjreddie.com/darknet/yolo/
[3] https://towardsdatascience.com/yolo-v3-explained-ff5b850390f.
[4] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229.
[5] https://medium.com/@anand_sonawane/yolo3-a-huge-improvement-2bc4e6fc44c5
[6] Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.

[7] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. International journal of computer vision, 115(3), 211-252.
[8] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
[9] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.
[10] Tolstikhin, I., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., ... & Dosovitskiy, A. (2021). Mlp-mixer: An all-mlp architecture for vision. arXiv preprint arXiv:2105.01601.
[11] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).
[12] Girshick, R. (2015). Fast r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 1440-1448).
[13] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28, 91-99.
[14] https://blog.paperspace.com/how-to-implement-a-yolo-object-detector-in-pytorch/
[15] https://towardsdatascience.com/breaking-down-mean-average-precision-map-ae462f623a52
[16] https://sandipanweb.wordpress.com/2018/03/11/autonomous-driving-car-detection-with-yolo-in-python